

CLAIMS

1 1. A method in a computer system for processing packets of a message, the
2 method comprising:
3 receiving a packet of the message;
4 identifying a component for processing the received packet;
5 receiving from the identified component an identifier of state information
6 associated with the message;
7 retrieving state information associated with the received identifier; and
8 providing the retrieved state information and the received packet to the
9 identified component for processing of the received packet.

1 2. The method of claim 1 including requesting that the identified
2 component provide an identifier of state information.

1 3. The method of claim 1 wherein the providing includes invoking a
2 message handler of the component.

1 4. The method of claim 1 wherein the receiving of the identifier is in
2 response to invoking a routine of the component.

1 5. The method of claim 1 wherein the component is a protocol.

1 6. A method in a computer system for processing packets of a message, the
2 method comprising:
3 receiving a packet of the message and a data type of the message;
4 identifying a component that is capable of processing a packet of the indicated
5 data type; and
6 providing the received packet to the identified component for processing.

1 7. The method of claim 6 including
2 receiving from the identified component an identifier of state information
3 associated with the message;
4 retrieving state information associated with the received identifier; and
5 providing the retrieved state information along with the received packet to the
6 identified component for processing.

1 8. The method of claim 6 wherein the receiving of the data type includes
2 requesting the data type from a component that previously processed the packet.

1 9. The method of claim 6 wherein the component is a protocol with an
2 edge.

1 10. A component in a computer system for message handling, the message
2 having packets, comprising:
3 for each of a plurality of processing sub-components,
4 a state key function for generating an identifier of state information
5 based on a packet; and
6 a message handler function for processing a packet of the message using
7 state information identified by the identifier; and
8 a session function for generating initial state information for a message that is
9 associated with a generated identifier; and

1 11. The component of claim 10 wherein the component is a protocol and the
2 sub-components are edges of the protocol.

1 12. The component of claim 10 wherein the message handler function
2 updates the state information.

1 13. The component of claim 10 wherein each sub-component is for
2 processing messages of different data types.

1 14. The component of claim 10 wherein multiple messages share the same
2 state information.

1 15. The component of claim 10 wherein multiple sub-components share the
2 same state information.

1 16. The component of claim 10 wherein the message handler function is
2 passed state information.

1 17. The component of claim 10 wherein the state information is stored
2 external to the component.

1 18. The component of claim 10 wherein the message handler converts data
2 of a packet.

1 19. A computer-readable medium containing a data structure comprising a
2 sequence of path entries, each path entry having a reference to state information for a
3 message and a reference to a message handler for processing a message wherein the message
4 handlers are to be invoked in the order of the sequence.

1 20. The computer-readable medium of claim 19 wherein the data structure
2 includes an indication of type of data to be output by the sequence of message handlers.

1 21. The computer-readable medium of claim 19 wherein a path entry
2 includes an indication of type of data output by the message handler.

1 22. A method in a computer system for processing a message, the message
2 having a plurality of headers, the method comprising:

3 analyzing the headers of the message to identify a sequence of message
4 handlers for processing the message; and

5 invoking some of the identified message handlers passing the message.

1 23. The method of claim 22 wherein the analyzing includes identifying a
2 data type associated with a header.

1 24. The method of claim 22 including locating state information based on
2 information in a header.

1 25. The method of claim 24 wherein the analyzing includes identifying a
2 state indicator routine for each message handler and the locating of state information
3 includes invoking the identified state indicator routine passing the message wherein the state
4 indicator routine advances a reference past the header associated with the state indicator
5 routine.

1 26. The method of claim 22 wherein the invoking is under control of a
2 single thread of execution.

1 27. The method of claim 22 wherein analyzing includes identifying multiple
2 sequences of message handlers.

1 28. The method of claim 22 wherein each invoked message handler
2 advances a reference past its header in the message.

1 29. A computer-readable medium that implements the method of claim 1.

1 30. A computer-readable medium that implements the method of claim 6.

1 31. A computer-readable medium containing a data structure that
2 includes:

3 a plurality of item fields, each item field identifying a conversion routine
4 for processing a message in sequence; and

5 a type field specifying that each item field contains the identifier of a
6 conversion routine.

1 32. The computer-readable medium of claim 31 wherein an item field
2 identifies a protocol and edge.

1 33. The computer-readable medium of claim 31 wherein an item field
2 includes an identifier of state information for the conversion routine.

1 34. The computer-readable medium of claim 31 wherein the data
2 structure is a URL.

~~Add A1~~

[illegible]